

## Assignment- 4

Assignment Date

30 October 2022

Student Name

Lalithraju A N

Student Roll Number

212219060147

Maximum Marks

2 Marks

Question 1:

Pull an image from docker hub and run it in docker playground.

The screenshot displays the Docker Playground interface. On the left, a sidebar shows a clock at 03:57:32, a 'CLOSE SESSION' button, and an 'Instances' section with a search icon and a settings gear. Below this is an 'ADD NEW INSTANCE' button and a list of instances, including one with IP 192.168.0.8 and name 'root1'. The main area shows a terminal window for instance 'cddvksm0\_cddvkvm0qau000a07j5g'. The terminal output includes a warning about the sandbox environment, followed by the execution of 'docker pull hello-world' and 'docker run hello-world', both of which succeed. The terminal also shows the IP address 192.168.0.8 and a button to 'OPEN PORT'. The CPU usage is shown as 0.31%.

```
03:57:32
CLOSE SESSION

Instances

+ ADD NEW INSTANCE

192.168.0.8
root1

cddvksm0_cddvkvm0qau000a07j5g

IP: 192.168.0.8 OPEN PORT

Memory: 1.24% (49.52MB / 3.99GB) CPU: 0.31%

SSH
ssh ip:192-168-0-22-cddvksm0qau000a07j5g@directlabs.gig

DELETE EDITOR

WARNING!!!!
# This is a sandbox environment. Using personal credentials
# is HIGHLY discouraged. Any consequences of doing so are
# completely the user's responsibility.
# The PND team.

[root@localhost ~]# docker pull hello-world
Using default tag: latest
latest: Pulling from library/hello-world
24629730121e: Pull complete
Digest: sha256:c2899a77aef4a607a671a6e3e4d05c1a577c95a1a64f132a60011d5a7
Status: Downloaded newer image for hello-world:latest
docker.io/library/hello-world:latest
[root@localhost ~]# docker run hello-world
```

03:57:05

CLOSE SESSION

Instances 🔧 ⚙️

+ ADD NEW INSTANCE

192.168.0.8

node1

cddvkxm0\_cddvkvm0qau000a07j5g

IP

192.168.0.8

OPEN PORT

Memory

1.26% (50.45MB / 3.906GiB)

CPU

0.39%

SSH

ssh ip172-18-0-22-cddvkxm0qau000a07j5g@direct.labs.pln

DELETES

EDITOR

2. The Docker daemon pulled the "hello-world" image from the Docker Hub. (amd64)

3. The Docker daemon created a new container from that image which runs the executable that produces the output you are currently reading.

4. The Docker daemon streamed that output to the Docker client, which sent it to your terminal.

To try something more ambitious, you can run an Ubuntu container with:

```
$ docker run -it ubuntu bash
```

Share images, automate workflows, and more with a free Docker ID: <https://hub.docker.com/>

For more examples and ideas, visit: <https://docs.docker.com/get-started/>

```
[root@ip172-18-0-22-cddvkxm0qau000a07j5g ~]# docker run --rm -it ubuntu bash
```

Activate Windows

Go to Settings to activate Windows.

Question 2:

Create a docker file for the job portal application and deploy it in Docker

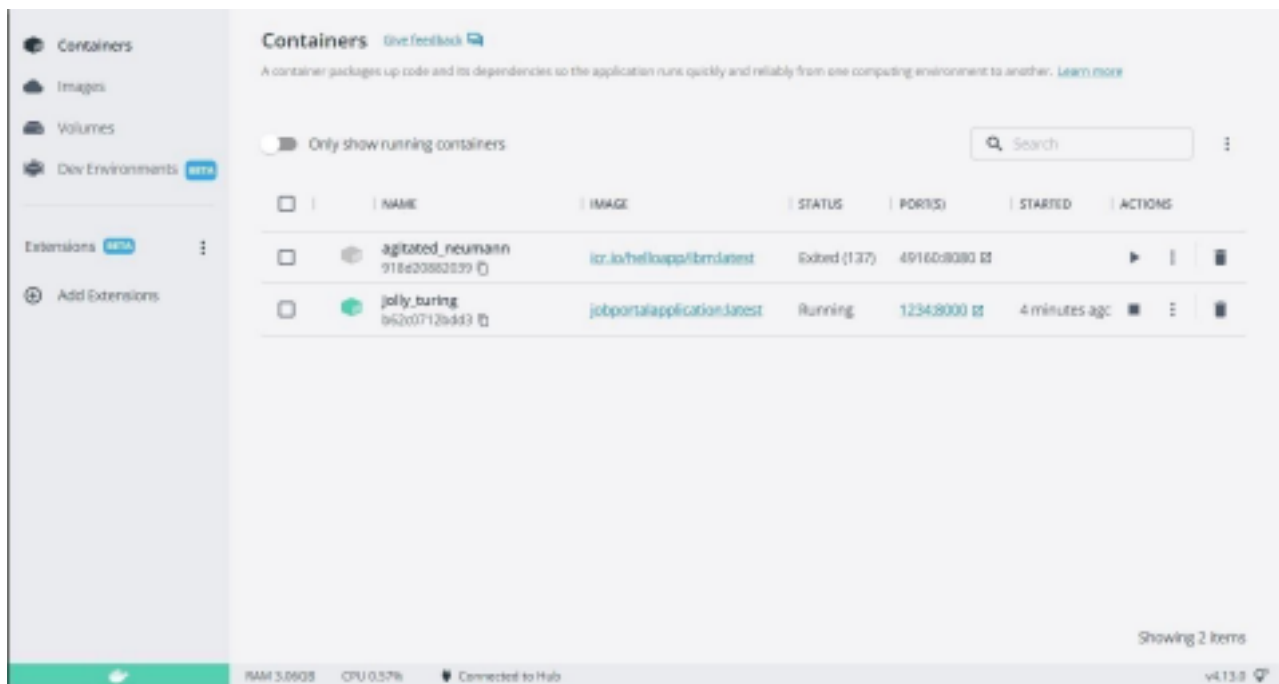
desktopapplication.DOCKER FILE:

```

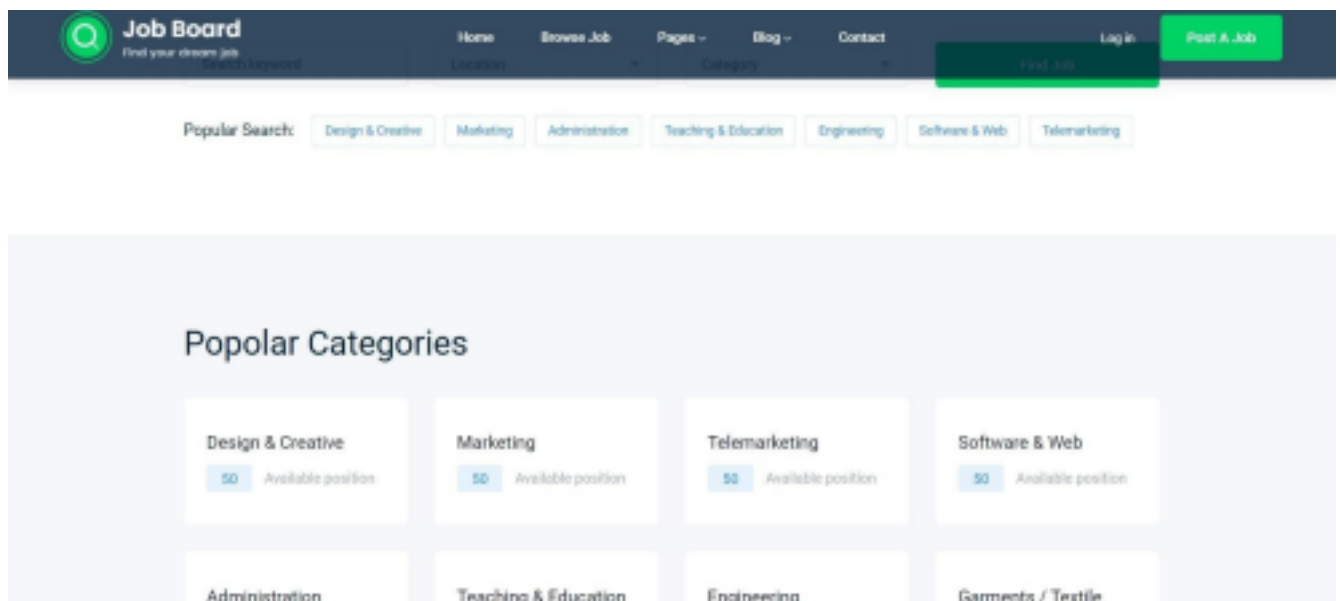
1 FROM python:3.8-buster
2
3 WORKDIR /app
4
5 COPY requirements.txt /app/
6
7 RUN pip install -r requirements.txt
8
9 COPY . /app/
10
11 RUN cp .env.dev.sample .env
12
13 EXPOSE 8000
14
15 RUN chmod +x entrypoint.sh
16
17 CMD ["sh", "entrypoint.sh"]

```

DEPLOYMENT OF JOBPOTAL APPLICATION:



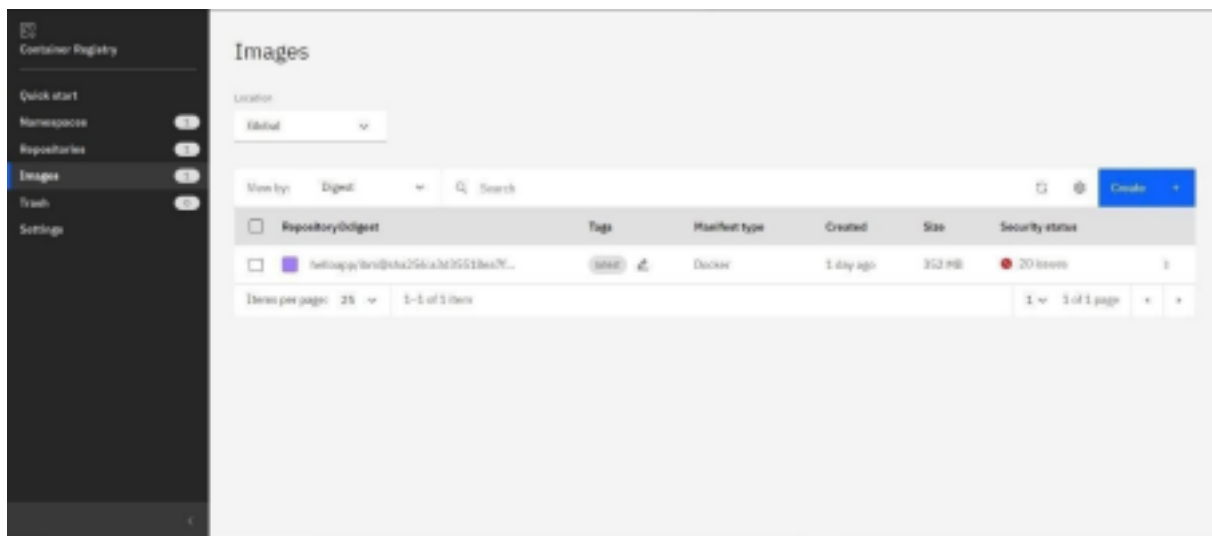
OUTPUT:



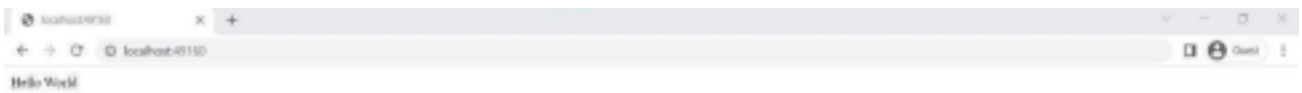
Question 3:

Create a IBM container registry and deploy hello-world app or job port app.IBM

CONTAINER REGISTRY DEPLOYMENT:



OUTPUT:



Question 4:

Create a Kubernetes cluster in IBM cloud and deploy hello world image or job portal image and also expose the same app to run in node port.

Creating Kubernetes cluster in IBM cloud and exposing node port:

